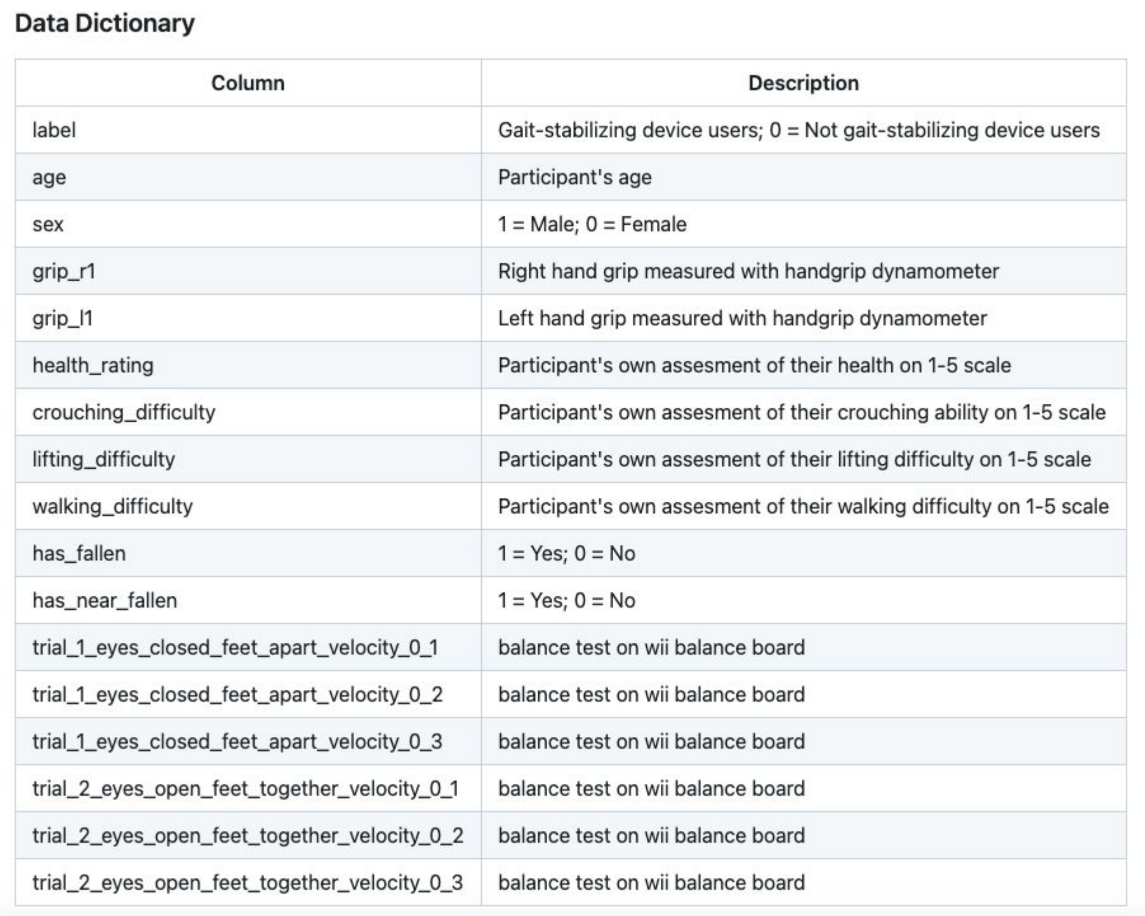
Markers for Detecting Falls in the Elderly

Timothy Peng
Data Science Capstone Project
Brainstation

Problem Statement 🔍

- One in four elderly suffers from a fall (CDC)
- Falling once doubles your chance of falling again
- Elderly (65+) account for 42% of the total healthcare spend
    - Only represent 17% of the population

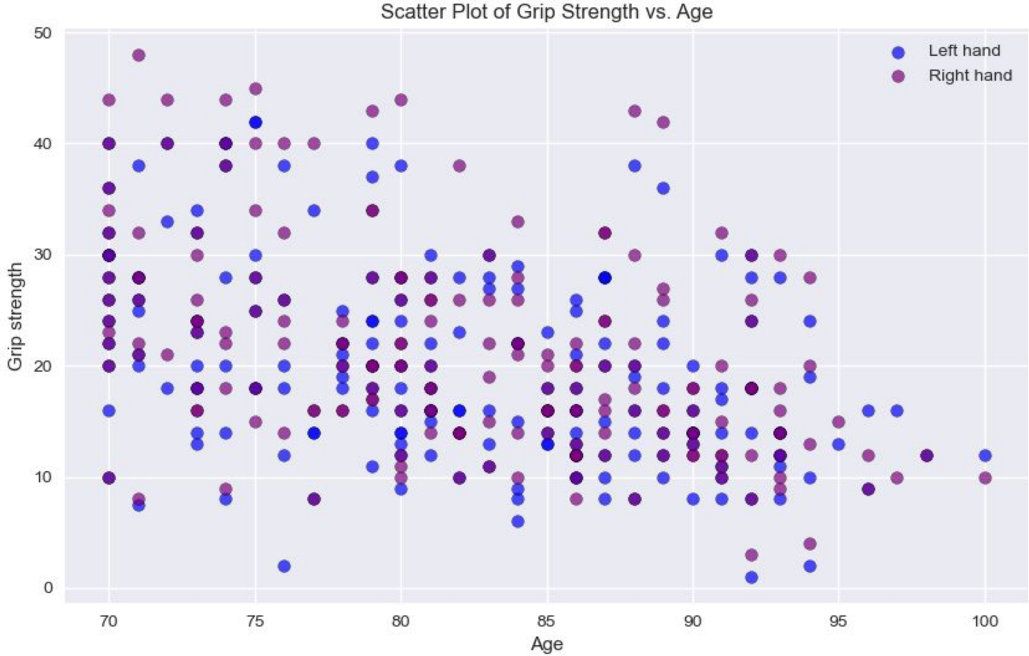
The Data 📊



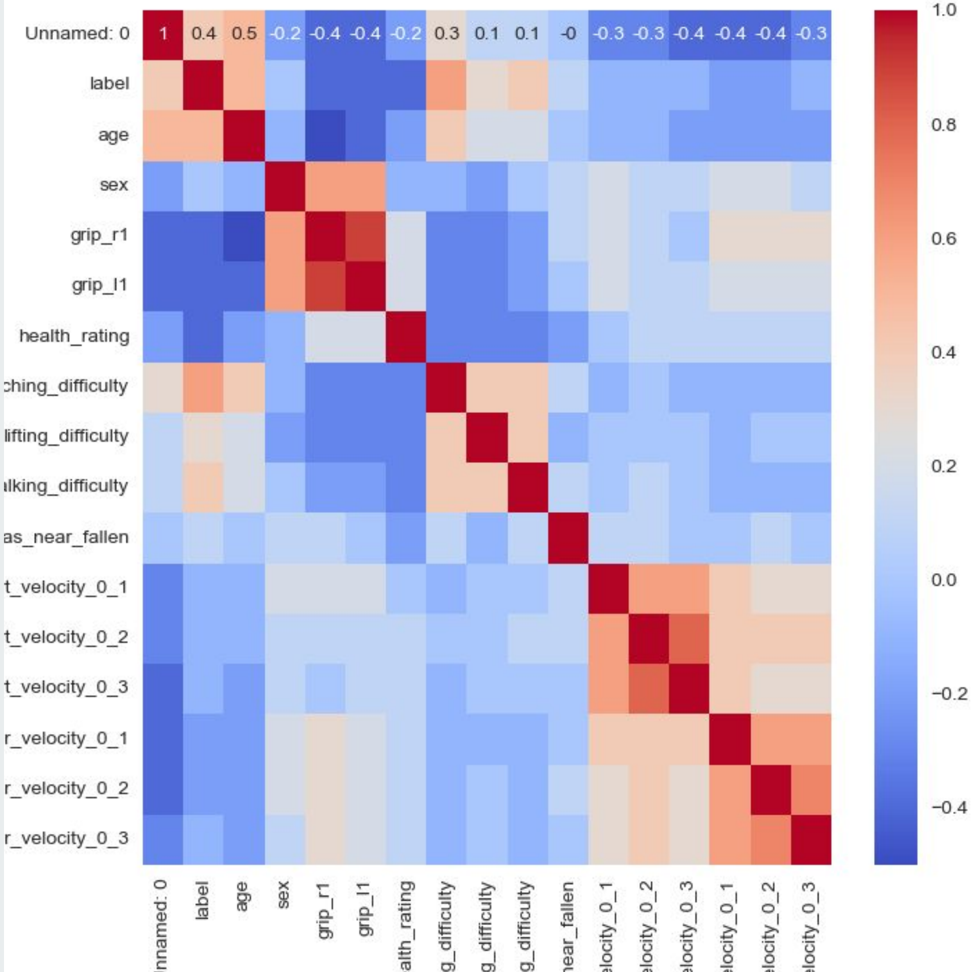
Data Dictionary

Column	Description
label	Gait-stabilizing device users; 0 = Not gait-stabilizing device users
age	Participant's age
sex	1 = Male; 0 = Female
grip_r1	Right hand grip measured with handgrip dynamometer
grip_l1	Left hand grip measured with handgrip dynamometer
health_rating	Participant's own assesment of their health on 1-5 scale
crouching_difficulty	Participant's own assesment of their crouching ability on 1-5 scale
lifting_difficulty	Participant's own assesment of their lifting difficulty on 1-5 scale
walking_difficulty	Participant's own assesment of their walking difficulty on 1-5 scale
has_fallen	1 = Yes; 0 = No
has_near_fallen	1 = Yes; 0 = No
trial_1_eyes_closed_feet_apart_velocity_0_1	balance test on wii balance board
trial_1_eyes_closed_feet_apart_velocity_0_2	balance test on wii balance board
trial_1_eyes_closed_feet_apart_velocity_0_3	balance test on wii balance board
trial_2_eyes_open_feet_together_velocity_0_1	balance test on wii balance board
trial_2_eyes_open_feet_together_velocity_0_2	balance test on wii balance board
trial_2_eyes_open_feet_together_velocity_0_3	balance test on wii balance board

The Data 📊


Scatter Plot of Grip Strength vs. Age
Grip strength
Age
0
10
20
30
40
50
70
75
80
85
90
95
100
Left hand
Right hand

Collinearity 🔗



Baseline models 📈

Model	Accuracy	Precision	Recall	F1
Logistic Regression	75%	67%	71%	69%
Random Forest Classifier	78%	55%	86%	67%

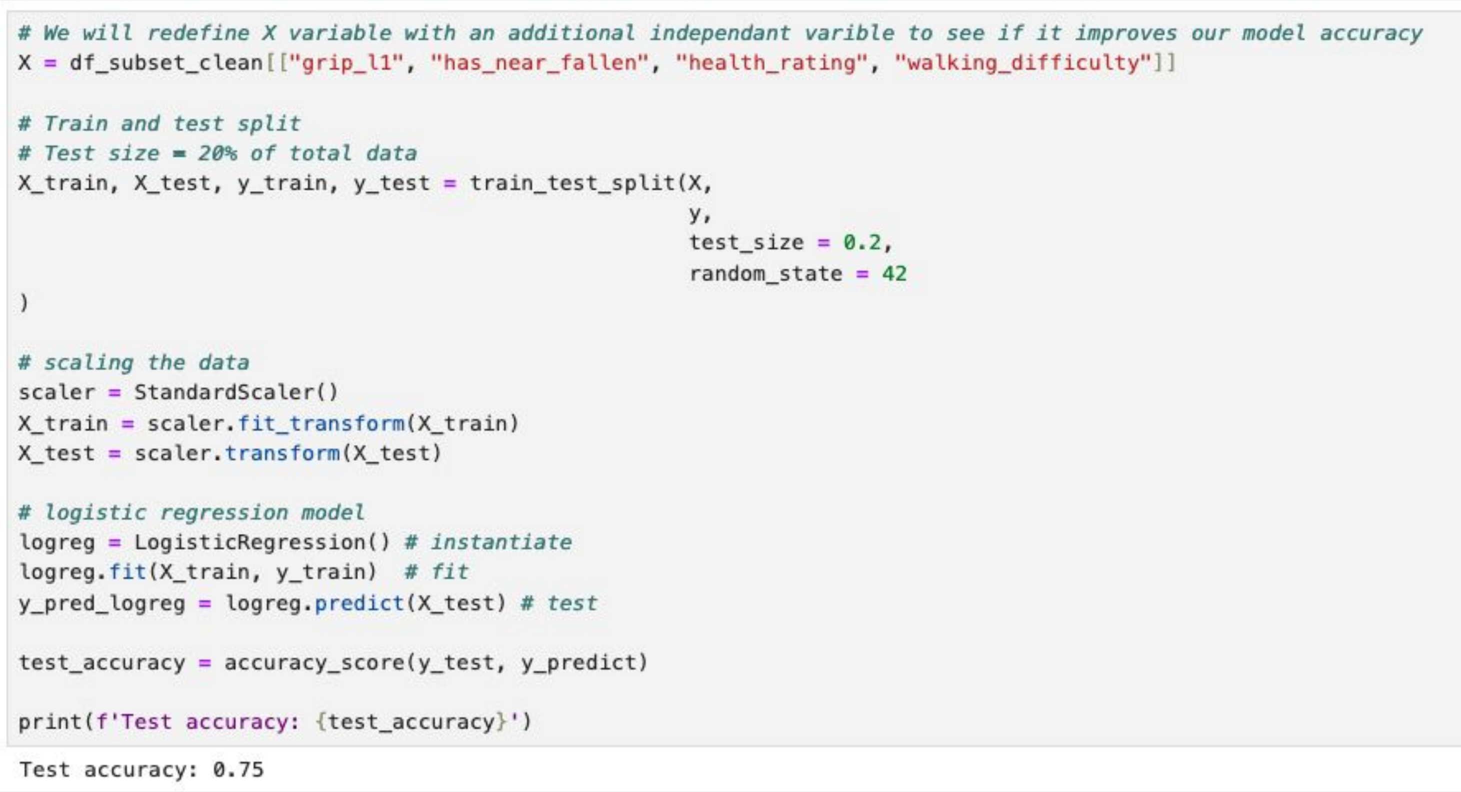
Next Steps 💡

- Model optimization
- Interpretation
- Final demonstration

Thanks for listening 🥳
Questions? 🙋‍♂️

Appendix 📑

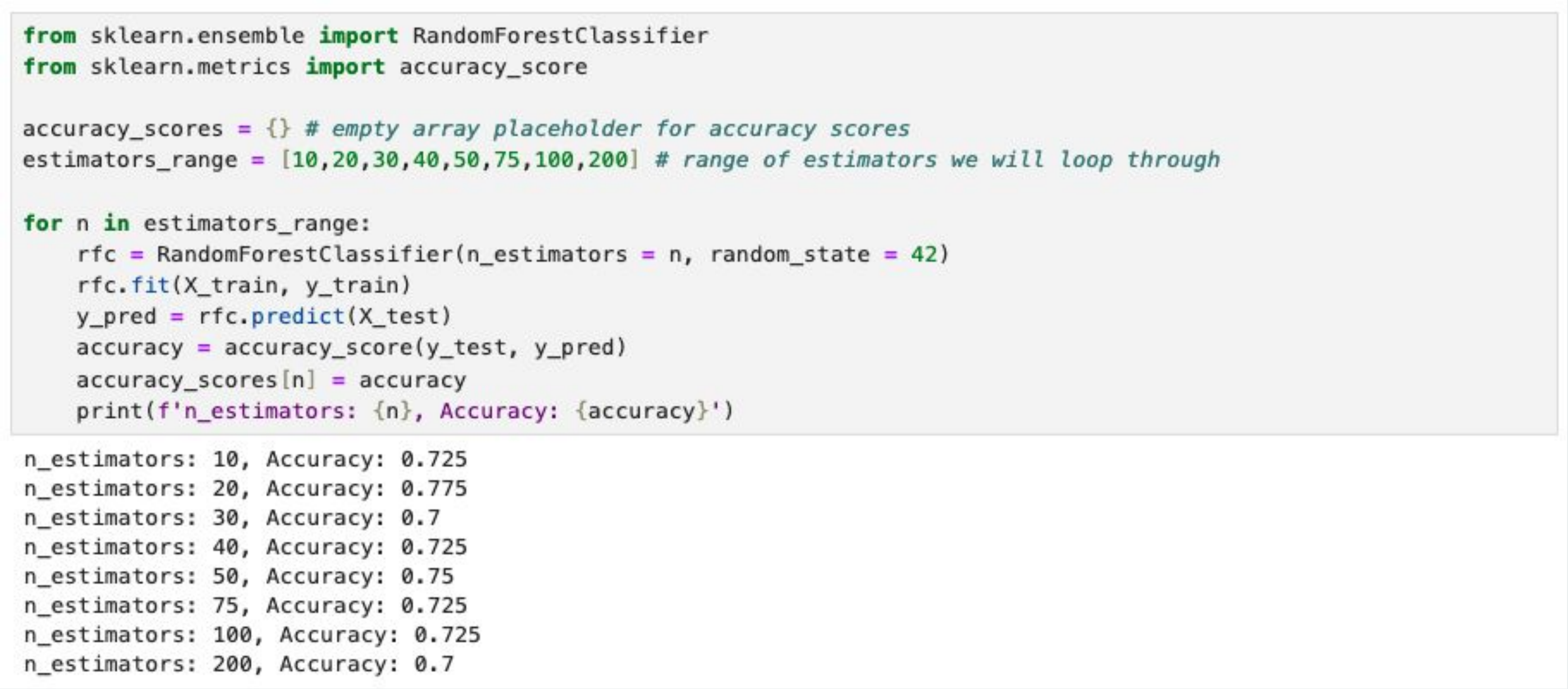
Logistic Regression



```
# We will redefine X variable with an additional independant varible to see if it improves our model accuracy
X = df_subset_clean[["grip_l1", "has_near_fallen", "health_rating", "walking_difficulty"]]

# Train and test split
# Test size = 20% of total data
X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size = 0.2,
                                                    random_state = 42
)

# scaling the data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# logistic regression model
logreg = LogisticRegression() # instantiate
logreg.fit(X_train, y_train)  # fit
y_pred_logreg = logreg.predict(X_test) # test

test_accuracy = accuracy_score(y_test, y_predict)

print(f'Test accuracy: {test_accuracy}')
```

```
Test accuracy: 0.75
```

Random Forest



```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

accuracy_scores = {} # empty array placeholder for accuracy scores
estimators_range = [10,20,30,40,50,75,100,200] # range of estimators we will loop through

for n in estimators_range:
    rfc = RandomForestClassifier(n_estimators = n, random_state = 42)
    rfc.fit(X_train, y_train)
    y_pred = rfc.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    accuracy_scores[n] = accuracy
    print(f'n_estimators: {n}, Accuracy: {accuracy}')
```

```
n_estimators: 10, Accuracy: 0.725
n_estimators: 20, Accuracy: 0.775
n_estimators: 30, Accuracy: 0.7
n_estimators: 40, Accuracy: 0.725
n_estimators: 50, Accuracy: 0.75
n_estimators: 75, Accuracy: 0.725
n_estimators: 100, Accuracy: 0.725
n_estimators: 200, Accuracy: 0.7
```